REMARKS

Claims 28-39 are pending in the present application. Claims 28-39 have been rejected. Claim 35 has been amended. No new matter has been added. Accordingly, claims 28-39 are now pending in the present application. Claim 28 has been amended to further facilitate moving prosecution forward. Additionally, Applicant has amended claim 35 to facilitate expeditious prosecution. Applicant respectfully reserves the right to pursue claims, including the subject matter encompassed by claims 28 and 35 as presented prior to this Amendment and additional claims in one or more continuing applications.

Claim Rejections - 35 USC § 112

Claim 35 was rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement. Claim 35 has been amended. Applicant believes the rejection is overcome in view of the amendment, where basis for the amendment may be found throughout the Specification, Figures and originally-filed claims, and more particularly at page 13, line 20 through page 14, line 17 of the Specification. Applicant respectfully requests removal of the rejection with respect to claim 35.

Claim Rejections - 35 USC § 103

Claims 28-33, 37, 39 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Hebert et al. (U.S. Patent No. 6,134,618) in view of Yao et al. (U.S. PGPUB No. 2003/0126280). Claims 34-36 and 38 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Hebert-Yao et al. further in view of Jarvis et al. (U.S. Patent No. 5,870,561). Applicant respectfully disagrees with the rejections.

Claim 28

Claim 28 is set forth below for convenience:

A system for managing congestion and avoidance behavior of network processors, the system comprising:

a plurality of network processors controlling network traffic, a first of the plurality of network processors being of a different model or version from a second of the plurality of network processors;

a host processor including a congestion control application that manages congestion and avoidance behavior of the plurality of network processors, the congestion control application being network processor independent such that the congestion control application need not have specific knowledge of a network processor's hardware, software, or firmware in order to manage the network processor's congestion and avoidance behavior;

a plurality of application programming interfaces (APIs), each of the plurality of APIs being usable by the congestion control application of the host processor, via a multi-word header of a plurality of words, to manage the congestion and avoidance behavior of any of the plurality of network processors, none of the plurality of APIs being limited for use with a specific network processor model or version, and

the multi-word header usable for congestion control, wherein the multi-word header is comprised of a plurality of words where a first part of the multi-word header consisting of a first plurality of words is common to the congestion control application and a second part of the multi-word header consisting of a second plurality of words is common to a plurality of congestion algorithms.

Reference of Herbert

Examiner has stated:

Regarding Claim 28, Hebert discloses a system for managing behavior of network processors, the system comprising:

a) a first of the network processors (Hebert col 2, ll 63-67; telecommunications switch (network processor) being of a different model or version from a second of the plurality of network processors; (Hebert col 3, ll 26-30; col 3, ll 46-55; universal (generic or non specific) API; col 3, l 61 - col 4, l 2; supports multiple protocol specific state machines; host to switch interface unchanged)

b) a host processor (Hebert col 2, ll 56-63: host to switch interface for controlling telecommunications devices (network processors) application that manages network processors, the network processor is independent (Hebert col 3, ll 26-30; col 3, ll 46-55: generic API; no specific telecommunications switch (network processor); col 3, l 61 - col 4, l 2: supports multiple protocol specific state machines; host to switch interface unchanged) such that the application need not have specific knowledge of a network processor's hardware, software, or firmware in order to manage the network processor's behavior; (Hebert col 3, ll 39-45: standardized interface for application development) and

c) a plurality of application programming interfaces (APIs) (Heber col 1, ll 22-27: multiple telecommunications switches; col 2, ll 56-63: host to switch API interface (multiple APIs)), each of the plurality of APIs being usable by the host processor to manage any of the plurality of network processors, none of the plurality of APIs being limited for use with a specific network processor model or version. (Hebert col 3, l 61 - col 4, l 2: supports multiple protocol specific state machines; host to switch interface unchanged; not limited to a specific telecommunications switch or network processor)

Applicant respectfully disagrees for the reasons following and requests removal of the rejections and reconsideration of the pending claims.

<u>The "telecommunications switch" of Hebert is</u>

<u>not a network processor of the present invention</u>

Applicant notes that Examiner did not address the claim portions of claim 28 of "A system for managing congestion and avoidance behavior of network processors, the system comprising: a plurality of network processors controlling network traffic," in the rejection. Applicant has reviewed Hebert and concludes that these claim portions were not addressed as Hebert does not provide for or teach towards "a system for managing congestion and avoidance behavior of network processors." Further Applicant concludes that Hebert does not provide for or teach towards "a plurality of network processors controlling network traffic." Applicant posits that this result is not unexpected as the Hebert reference fails to provide for such as the Hebert reference is not an approach for such, contrary to the present invention. Rather Hebert is suited for a **call control processing using a universal API through switched message traffic** and not a system for managing congestion and avoidance behavior of network processors in accordance with the present invention. Applicant respectfully submits that Hebert does not render the present invention unpatentable.

Hebert is directed to a "universal host-to-switch application program interface (API) utilizing a generic message format … to meet … network signalling protocol requirements" in which data and commands are transmitted "between the host application and the switch." (Abstract). Hebert then seeks to provide a "**universal API that provides standardized call control processing** by utilizing one or more generic message formats and **supporting host-to-switch call control processing** that may be used regardless of the host application or signalling protocol requirements." (Column 2, lines 44-48, emphasis added). Hebert uses a switch to control telecommunications services, where the Herbert switch is "used to control or manage a wide variety of communications services within **a programmable switch through the transfer of the generic API messages**, including conferencing, voice recorded announcements, tone generation, tone reception, call progress analysis, voice recognition, voice compression and fax encoding/decoding." (Column 3, lines 20-25, emphasis added). The switch of Hebert then is not a network processor as in the present invention.

Hebert sets forth a "programmable telecommunication switch that provides a user with the ability to define a desired signalling protocol, either 'standard' or custom in nature, **for performing any desired switching functions**" in order to provide the transmitted messages in accordance with the signaling protocol requirements of the network. (Abstract) Hebert uses a switch to control telecommunications services using switching functions of the switch with regard to the Hebert approach. The switch of Hebert does not perform network processing but rather is "usually **controlled by a host device**, which is typically a computer that runs a telecommunications application program" (Column 1, lines 24-27); "**resides within** a PC 102" (Column 5, line 27); "can reside on a passive backplane (no PC CPU 104 or disk 106 present) from which it receives electrical power and **be controlled by the external host** 130" (Column 5, lines 60-63); does not perform processing related to or in substitute for a PC CPU 104 as it does not have the capability to perform processing as such; and is used with the universal APIs of Hebert **to control communications services "through the transfer of the generic API messages**" (Column 3, lines 20-22).

Further Examiner has cited Hebert at Column 2, lines 63-67 as providing support for Examiner's assertion that the "telecommunications switch" of Hebert is a first of the network processors. Applicant respectfully disagrees and is unable to find supporting basis or prima facie evidence for the assertion in the cited portion of Hebert; Applicant has set forth the cited section as:

"The present invention further comprises a programmable telecommunication switch that provides a user with the ability to define a desired signalling protocol, either "standard" or custom in nature, for performing any desired switching functions."

Further Examiner's rejections based on Hebert's allegedly disclosing "being of a different model or version from a second of the plurality of network processors" is also mischaracterized. Hebert sets forth in the cited portions: "For example, the generic messages of the universal API may be used to support communications between any software layer within the switch." Applicant notes that the

universal API of Hebert is not a network processor, and message traffic support within the switch does not constitute network processing. Hence, there is no network processor disclosed and there is no different model or version from a second of the plurality of network processors disclosed in Hebert.

Additionally, at the cited portions, Hebert provides:

"Another advantage of the present invention is that it enables a user to create multiple network signalling protocols by creating separate state machines to address each variation of a signalling protocol. **The universal API may be programmed** to achieve the necessary communications **to support each of these protocol-specific state machines**. Thus, the structure of the messages comprising the host-to-switch interface may remain unchanged despite the **multiple signalling protocols supported by the switch**."

Applicant respectfully disagrees with Examiner's assertion and notes this cited portion again provides no supporting basis or prima facie evidence for the rejection.

Hence, for each and all of the reasons above, Applicant respectfully disagrees with the Examiner's assertion that the "telecommunications switch" of Hebert is a network processor. Therefore, Hebert does not disclose, teach, suggest or otherwise motivate one towards: "a system for managing congestion and avoidance behavior of network processors, the system comprising: a plurality of network processors controlling network traffic, a first of the plurality of network processors being of a different model or version from a second of the plurality of network processors" as recited in claim 28.

Hebert does not

disclose a host processor of the present invention

Examiner has stated that Hebert discloses: "a host processor including a congestion control application that manages congestion and avoidance behavior of the plurality of network processors, the congestion control application being network processor independent such that the congestion control application need not have specific knowledge of a network processor's hardware, software, or firmware in order to manage the network processor's congestion and avoidance behavior."

Applicant respectfully disagrees. Applicant incorporates the reasoning above herein and further presents additional remarks founding and evidencing Applicant's position. Examiner has cited portions of Hebert including:

"The present invention is **a universal host-to-switch application program interface (API)** utilizing a generic message format for performing call control processing and capable of being customized to meet telecommunications application and network signalling protocol requirements. The generic message formats have programmable fields for transmitting commands, status, and data between the host application and the switch. The present invention **further comprises a programmable telecommunication switch that provides a user with the ability to define a desired signalling protocol, either "standard" or custom in nature, for performing any desired switching functions**."

"Another advantage of the **generic message structure** of the present invention is that it provides the commonality and flexibility necessary to be a **standardized interface** for application development. This significantly reduces the complexity of the host/switch communications interface and eliminates the cost of supporting an interface composed of numerous specialized messages."

As reasoned before, Hebert is <u>not</u> a network processor and is <u>not</u> a host processor. Hebert is an interface designed to be a standard interface for specific approaches of Hebert.

Hence, for each and all of the reasons above, Applicant respectfully disagrees with the Examiner's assertion and submits that Hebert does not disclose, teach, suggest or otherwise motivate one towards: "a host processor including a congestion control application that manages congestion and avoidance behavior of the plurality of network processors, the congestion control application being network processor independent such that the congestion control application need not have specific knowledge of a network processor's hardware, software, or firmware in order to manage the network processor's congestion and avoidance behavior" as recited in claim 28.

<u>Hebert does not</u>

<u>disclose a plurality of APIs of the present invention</u>

Examiner has stated that Hebert discloses: "a plurality of application programming interfaces (APIs), each of the plurality of APIs being usable by the congestion control application of the host processor to manage the congestion and avoidance behavior of any of the plurality of network

processors, none of the plurality of APIs being limited for use with a specific network processor model or version."

Applicant respectfully disagrees. Applicant incorporates the reasoning above herein and further presents additional remarks founding and evidencing Applicant's position. Examiner has cited portions of Hebert including:

"Programmable telecommunication switches are used in a wide variety of applications such as voice messaging, telemarketing services and the like. program **A programmable switch is usually controlled by a host device**, which is typically a computer that runs a telecommunications application. A customer may either purchase a commercially available application program that is compatible with the host and switch hardware or may elect to write a custom program."

"The present invention is **a universal host-to-switch application program interface (API)** utilizing a generic message format for performing call control processing and capable of being customized to meet telecommunications application and network signalling protocol requirements. The generic message formats have programmable fields for transmitting commands, status, and data between the host application and the switch."

"Another advantage of the present invention is that it enables a user to **create multiple network signalling protocols** by creating separate state machines to address each variation of a signalling protocol. **The universal API may be programmed to achieve the necessary communications to support each of these protocol-specific state machines.** Thus, the structure of the messages comprising the host-to-switch interface may remain unchanged **despite the multiple signalling protocols supported by the switch.**"

As reasoned before, Hebert is not a network processor and is not a host processor and does not provide for a plurality of APIs. Hebert is a single universal interface designed to be a standard interface for specific approaches of Hebert. Hebert provides for creating multiple signaling protocols using a single universal API so the host-to-switch interface may remain unchanged, but does not provide for a plurality of APIs.

Hence, for each and all of the reasons above, Applicant respectfully disagrees with the Examiner's assertion and submits that Hebert does not disclose, teach, suggest or otherwise motivate one towards: "a plurality of application programming interfaces (APIs), each of the plurality of APIs being usable by the congestion control application of the host processor to manage the congestion and

avoidance behavior of any of the plurality of network processors, none of the plurality of APIs being limited for use with a specific network processor model or version" as recited in claim 28.

Reference of Yao

Examiner has stated that:

Yao discloses for a): a plurality of network processors controlling network traffic; for b): a congestion control application; the plurality of network processors for c): congestion control application. (Yao Figure 1 (30); para 005, ll 1-4: multiple network processors providing flow control; para 006, ll 1-11; para 007, ll 29-45: XOFF message processed when high watermark reached; XON message processed when low watermark reached; congestion control method)

It would have been obvious to one of ordinary skill in the art to modify Hebert to use a congestion control application as taught by Yao. One of ordinary skill in the art would have been motivated to employ the teachings of Yao in order to use a XON/XOFF flow control scheme that prevents problems caused by HOL blocking such as increased system latency, unintentionally dropped packets, and time-out problems. (Yao para 030, ll 10-14: " … *The XON/XOFF flow control scheme prevents problems caused by HOL blocking such as increased system latency, unintentionally dropped packets, and time-out problems. Other advantages will be apparent to one of ordinary skill in the pertinent arts.* … ")

Applicant respectfully disagrees.

Yao is directed to "[a] system and method for providing XON/XOFF port-level flow control for a computer network that has access to a plurality of network processors in communication with the computer network" (Abstract). Although Yao discloses network processors, Yao does not disclose, teach, or suggest "a host processor including a congestion control application that manages congestion and avoidance behavior of the plurality of network processors" and "a plurality of application programming interfaces (APIs) . . . usable by the congestion control application of the host processor to manage the congestion and avoidance behavior of any of the plurality of network processors".

More particularly, Examiner has cited [005] as being a disclosure of multiple network processors providing flow control. However, upon close inspection of [005], the following is directly stated in Yao:

"The present invention is directed to a method for providing XON/XOFF port-level flow control for a computer network that has access to a plurality of network processors in communication with the computer network. At least one network processor has an egress port associated with an egress buffer, and a set of network processors is associated with a bridge. "

Yao does not disclose, teach, or suggest any "host processor [that] include[es] a congestion control application that manages congestion and avoidance behavior of the plurality of network processors", as recited in claim 28. In fact, Yao specifically teaches that each network processor monitors its own buffers for congestion and sends its own XON/XOFF flow control messages when applicable (see, e.g., FIG. 2 of Yao and corresponding description). Additionally, Yao does not disclose, teach, or suggest any "application programming interfaces (APIs) . . . [where] none of the . . . APIs [is] limited for use with a specific network processor model or version", as recited in claim 28. As clearly illustrated in FIG. 1 of Yao, each "network processor 30" in Yao has its own "interface 40".

Hence, for each and all of the reasons above, Applicant respectfully disagrees with the Examiner's assertion and submits that Yao does not disclose, teach, suggest or otherwise motivate one towards elements of the present invention as claimed.

Reference of Jarvis

Examiner has rejected Claims 34-36 and 38 under 35 U.S.C. § 103(a) as being unpatentable over Hebert-Yao et al. further in view of Jarvis et al. (U.S. Patent No. 5,870,561). Applicant incorporates the discussions above herein and further notes that Jarvis is directed to "[a] policy-driven network traffic manager [that] recommends to individual application programs that generate network traffic whether, and optionally under what conditions, they should generate network traffic" (Abstract of Jarvis). Jarvis does not disclose, teach, or suggest "a plurality of network processors controlling network traffic", "a host processor including a congestion control application that manages congestion and avoidance behavior of the plurality of network processors", and "a plurality of application programming interfaces (APIs) . . . usable by the congestion control application of the host processor to manage the congestion and avoidance behavior of any of the plurality of network processors", as recited in claim 28, nor may the "application programs" in Jarvis be construed as disclosing, teaching, or suggesting the "network processors" because programs are not processors. In addition, the "application programs" in Jarvis do not control network traffic. Rather, the "application programs" generate network traffic. Further Jarvis does not disclose, teach, or suggest a plurality of network processors as in claim 38.

Further, the "network traffic manager" in Jarvis cannot be construed as disclosing, teaching, or suggesting the "host processor" recited in claim 28 because the "network traffic manager" in Jarvis is simply a program, not a processor. Moreover, the "network traffic manager" in Jarvis only manages the

behavior of "application programs", which cannot be construed as disclosing, teaching, or suggesting the "network processors" recited in claim 28.

Although Jarvis discloses "an application programming interface (API) 210, by which the clients 200 . . . make requests 212 and receive recommendations 214 as to whether the clients should generate the proposed network traffic" (col. 5, lns. 1-5 of Jarvis), the "API" in Jarvis cannot be construed as disclosing the "APIs" recited in claims 28, and 34-36 because the "API" in Jarvis is not usable to manage "network processors", which Jarvis does not disclose, teach, or suggest.

Therefore, Jarvis does not disclose, teach, or suggest the elements of the present invention, in any claims.

Additional Comments Concerning Claim 28

Applicant further asserts that Claim 28, as amended traverses the rejections set forth. Although Applicant believes the arguments set forth above are persuasive, Applicant has further amended claim 28 to move prosecution forward.

Accordingly, and more particularly, Applicant asserts that none of the references cited anticipate, teach or otherwise disclose, alone or in combination "a plurality of application programming interfaces (APIs), each of the plurality of APIs being usable by the congestion control application of the host processor, via a multi-word header of a plurality of words, to manage the congestion and avoidance behavior of any of the plurality of network processors, none of the plurality of APIs being limited for use with a specific network processor model or version, and

the multi-word header usable for congestion control, wherein the multi-word header is comprised of a plurality of words where a first part of the multi-word header consisting of a first plurality of words is common to the congestion control application and a second part of the multi-word header consisting of a second plurality of words is common to a plurality of congestion algorithms."

Applicant therefore requests removal of all rejections and a timely issuance of pending claims.

CONCLUSION

Thus, neither Hebert nor Yao nor Jarvis, alone or in combination, disclose, teach, or suggest "a host processor including a congestion control application that manages congestion and avoidance behavior of the plurality of network processors" and "a plurality of application programming interfaces (APIs) . . . usable by the congestion control application of the host processor to manage the congestion and avoidance behavior of any of the plurality of network processors", as recited in claim 28. Consequently, even if Hebert and Yao were combined, the combination would neither teach nor suggest the elements of claim 28 or any dependent claims. Similarly, even if Hebert and Yao were combined in view of Jarvis, the combination would neither teach nor suggest the elements of claim 34-36 or 38.

Accordingly, based at least on the reasons above, Applicant respectfully submits that claim 28, and the claims that depend therefrom, are patentable over Hebert, in view of Yao, and are patentable over Hebert and Yao in view of Jarvis.

On the basis of the above remarks, reconsideration and allowance of the claims is believed to be warranted and such action is respectfully requested. Applicant respectfully requests removal of all

rejections. If the Examiner has any questions or comments, the Examiner is respectfully requested to contact the undersigned at the number listed below.

Applicant respectfully requests that this Amendment be entered as the amendment is timely filed after the final rejection, places the application in condition for allowance and in better form for the purposes of appeal.

Respectfully submitted,
SAWYER LAW GROUP LLP

Dated: August 11, 2008

/Joseph A. Sawyer, Jr./
Joseph A. Sawyer, Jr.
Attorney for Applicant
Reg. No. 30,801
(650) 475-1435